Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Информационно-аналитические системы

Ефремовой Елизаветы Андреевны

Система аутентификации пользователей

Выпускная квалификационная работа

Научный руководитель:
д. т. н., профессор Крук Е. А.

Рецензент:
к.т. н., доцент Овчинников А. А.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Analytical Information Systems

Efremova Elizaveta

User authentication system

Graduation Thesis

Scientific supervisor:
Dr. Sci., professor Krouk Evgenii

Reviewer:
Cand. Sci., docent Ovchinnikov Andrei

Saint-Petersburg
2017

Оглавление

Введение

В последнее время облачные системы достигли беспрецедентного уровня использования. Несмотря на несомненные преимущества облака перед традиционными системами, к их недостаткам можно отнести риски, связанные с безопасностью передачи и приема данных. Таким образом встает вопрос о реализации надежной системы аутентификации.

На текущий момент существует несколько различных способов аутентификации, среди них – аутентификация с помощью пароля(одноразового, многоразового), токена или электронной подписи.[8]

В процессе реализации системы аутентификации в облаке зачастую необходимо убедиться в неотслеживаемости передаваемой аутентификационной информации. Среди различных способов решения данной проблемы существует метод использования гомоморфного шифрования при передаче аутентификационных данных облаку. Данное шифрование, помимо всего прочего, позволяет производить скрытные вычисления, т.е. не передавать в облако никаких реальных данных, а только зашифрованные, без возможности расшифрования на сервере, нивелируя тем самым большинство проблем связанных с безопасностью в облаке. Однако, реализация полностью гомоморфной системы, которую можно использовать на практике является сложной задачей и до последнего времени вообще считалась неразрешимой. Лишь после 2009 года, когда Крейг Джентри в своей работе доказал возможность существования подобной системы, начались активные исследования в этой области. В данной работе я рассматриваю один из вариантов реализации подобной системы.

1. Постановка задачи

Целью данной работы является написание библиотеки для работы с предложенной полностью гомоморфной системой шифрования. Для достижения данной цели были поставлены следующие задачи:

1. Изучение предметной области
2. Реализация библиотеки
3. Тестирование библиотеки
4. Анализ производительности системы
5. Криптоанализ

2. Схемы полностью гомоморфного шифрования

Начало исследований в данной области было положено в 1978 году, когда Рональд Линн Ривест, Леонард Адлеман и Михаил Дертузос опубликовали статью "О банках данных и гомоморфизмах, сохраняющих конфиденциальность" [5]. Рассматривая проблему сохранения конфиденциальности пользовательских данных в банке, ученые предложили как один из вариантов решения использовать специальный "privacy homomorphism", чтобы оперировать зашифрованными данными без промежуточного расшифрования и таким образом сэкономить на оборудовании и продолжить использовать ресурсы других фирм для хранения данных. Важным результатом данной статьи в истории гомоморфного шифрования стала открытая математическая задача построения защищенной полностью гомоморфной системы, эффективно реализуемой на практике.

Пусть A алгебраическая система, которая состоит из множества-носителя S, операций $f_1, f_2, ..$, предикатов $p_1, p_2, ..$ и некоторых выделенных констант $s_1, s_2, ...$ Будем обозначать такую систему следующим образом $A =< S; f_1, f_2, ...; p_1, p_2, ...; s_1, s_2, ... >$. Например, систему, состоящую из целых чисел с привычным набором операций, можно было бы обозначить так: $< Z; +, -, *, \div; <; 0, 1 >$.

Рассмотрим две алгебраические системы U и C, заданные как:

$$\begin{aligned} U &=< S; f_1, f_2, ...f_k; p_1, p_2, ...p_m; s_1, s_2, ...s_l > \\ C &=< S'; f'_1, f'_2, ...f'_k; p'_1, p'_2, ...p'_m; s'_1, s'_2, ...s'_l >, \end{aligned} \tag{1}$$

и обратимую функцию $\phi : S \to S'$.

Определение 1. (*privacy homomorphism*)

Рассматриваемая выше функция ϕ называется ***privacy homomorphism***,

если верны следующие условия:

$$\begin{aligned} f'_i(a,b,c,...) &= \phi^{-1}(f_i(\phi(a),\phi(b),\phi(c),...)), \forall i \in 1,...,k \\ p'_i(a,b,c,...) &= p_i(\phi(a),\phi(b),\phi(c),...), \forall i \in 1,...,m \\ \phi(s_i) &= s'_i, \forall i \in 1,...,l, \end{aligned} \tag{2}$$

а также:

1. Функции ϕ и ϕ^{-1} легко вычислимы;
2. Операции f'_i и p'_i также легко вычислимы;
3. Представление зашифрованного значения $\phi(a)$, где $a \in S$ требует не намного больше памяти, чем a;
4. Знания большого набора $\phi(a_i)$ недостаточно для восстановления ϕ;
5. Знания некоторых пар $\{a_i, \phi(a_i)\}$ недостаточно для восстановления ϕ;
6. Знания алгоритмов вычисления f'_i и p'_i недостаточно для восстановления ϕ.

Определение 2. (*гомоморфное шифрование*)

Положим, что для алгебраических систем U и C (1) можно построить ***гомоморфное шифрование***, если для них существует большое количество функций ϕ, которые являются privacy homomorphism, и количество операций в U и C больше нуля.

Определение 3. (*полное гомоморфное шифрование*)

Гомоморфное шифрование, в котором U и C - кольца, а среди операций f_i присутствуют операции сложения и умножения, называется ***полным гомоморфным шифрованием.***

Замечание 1 *Если же среди операций f_i присутствует только операция сложения или умножения, то тогда это* ***частично гомоморфное шифрование***

В 2009 году Крейг Джентри в своей кандидатской работе [2] ([3]) доказал возможность существования полностью гомоморфной системы. Он представил систему, основанную на теории целочисленных решеток. Основная идея заключалась в добавлении некоторого целочисленного шума к данным, без знания которого невозможно расшифровать результат. К сожалению, с ростом количества операций значение шума сильно возрастало и после определенного порога, восстановить результат было не возможно, но несмотря на это именно его доказательство существования таких систем стало причиной развития исследований в области защищенных вычислений.

В 2011 году исследователи из лаборатории НГУ-Parallels в своей работе "Безопасные облачные вычисления с помощью гомоморфной криптографии" [6], представили систему защищенных облачных вычислений, основанную на гомоморфизме в кольце многочленов от некоторой формальной переменной. Идея заключалась в том, что каждому числу $a_0 \in Z_n$ сопоставлялся многочлен вида $a(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_kx^k$, причем коэффициенты $a_i, i \in [1..k]$ и степень полинома k выбирались произвольно. Заметим, что если рассмотреть несколько целых чисел $\{a, b, c, ...\}$ и их полиномиальное представление $\{a(x), b(x), c(x), ...\}$, то $a+b+c+... = a(x)+b(x)+c(x)+...$ и $a*b*c*... = a(x)*b(x)*c(x)*...$. Таким образом, построенный гомоморфизм $G : Z_n \to Z_n[x]$ позволял производить защищенное сложение, вычитание и умножение чисел из Z_n.

Главное преимущество, которое получили исследователи НГУ-Parallels по сравнению с работой Джентри 2009 года заключается в отсутствии накапливающейся ошибки, и как следствие их схема имеет более простую математическую структуру данных, а значит повышает эффективность работы с зашифрованными данными. При всех преимуществах перед системой Крейга Джентри у данной схемы шифрования есть и ряд серьезных недостатков: неограниченный рост степеней многочленов приводит к тому, что несмотря на теоретическое отсутствие ограничений на количество возможных операций, на практике при большом числе требуемых умножений система становится достаточно быст-

ро вычислительно неэффективной; в системе не реализуемо деление, а также, сами данные представляются неэффективно, так как фактически требуется выполнять операции только над свободными членами полиномиальных представлений.

В 2015 году А. Абрамовым разработана полная гомомоморфная система [7], основанная на кольце многочленов с рациональными коэффициентами. Основное отличие от предыдущей работы заключается в возможности реализации операции деления, а также в ней предлагается более эффективное хранение данных.

3. Описание реализуемой криптосистемы

В данной работе реализована криптосистема, которая была предложена А. Абрамовым. Данная система имеет несколько преимуществ перед системой лаборатории НГУ-Parallels:

1. Более эффективное хранение данных;

2. Гомоморфизм относительно операции деления с остатком.

Несмотря на перечисленные преимущества, остается проблема неограниченного роста степеней, для решения которой необходимо научиться переходить к некоторым алгебраическим структурам с конечным числом элементов. В рамках данной работы, эта проблема остается нерешенной.

3.1. Гомоморфизм колец Z_n и $Z_n[x]$

Рассмотрим алгоритмы, необходимые для построения отображения Gom из кольца целых чисел Z_n в кольцо многочленов $Z_n[x]$.

Алгоритм 3.1. (*Преобразование числа в многочлен*)

1. Для каждого целого числа $a \in Z$ выбираем $r \in Z : r < a$ и переводим a в систему счисления с основанием r: $a_{10} = (a_m, a_{m-1}.., a_0)_r$;

2. Полученное представление числа a в системе счисления с основанием r перезаписываем в виде $a(x) = a_0 + a_1x + a_2x^2 + ... + a_mx^m$;

3. Выбираем произвольный многочлен $c(x)$, учитывая, что уравнение $c(x) = r$ должно иметь единственный действительный корень в кольце $Z(x)$: $\exists! g : c(g) = r$;

4. Результат композиции $res(x) = a(c(x))$ и есть представление выбранного числа a в кольце $Z(x)$.

Замечание 2 *Добавление к алгоритму 3.1: при преобразовании целочисленного вектора* $\{a_1, a_2, ..., a_m\}, n \in Z$ *параметры* r *и* $c(x)$ *являются общими, при этом* $r < max\{a_i, i \in [1..m]\}$.

Алгоритм 3.2. (*Обратный алгоритм*)

1. Для обратного преобразования нам необходимо знать r и $c(x)$ из алгоритма 3.1.

2. Результатом будет $res(x_r)$, где x_r - корень $s(x) = r$.

Построим отображение Gom из кольца целых чисел Z_n в кольцо многочленов относительно одной формальной переменной $Z_n[x]$ с помощью алгоритма 3.1:

$$Gom : Z_n \to Z_n[x] \tag{3}$$

Теорема 1.

Отображение Gom является гомоморфизмом колец Z_n и $Z_n[x]$.

Доказательство:

По определению гомоморфизма колец необходимо показать, что выполняются условия:

1. $Gom(a +_Z b) = Gom(a) +_{Z[x]} Gom(b)$

2. $Gom(a *_Z b) = Gom(a) *_{Z[x]} Gom(b)$,

для произвольных $a, b \in Z$.

Выбираем основание системы счисления $r : r < \{a, b\}$, тогда представление a и b - $a(x) = a_m x^m + a_{m-1}x^{m-1} + ... + ax + a_0$ и $b(x) = b_k x^k + b_{k-1}x^{k-1} + ... + bx + b_0$ в $Z[x]$ соответственно, где $a_{10} = (a_m, a_{m-1}, ..., a_0)_r$ и $b_{10} = (b_k, b_{k-1}, ..., b_0)_r$. НУО положим, что $m > k$.

1. $Gom(a)+_{Z[x]}Gom(b) = a(x)+b(x) = a_m x^m+...+(a_k+b_k)x^k+...+(a_0+b_0) = Gom(a +_Z b)$

2. Аналогично сложению. **Чтд.**

Таким образом, рассматриваемая система А. Абрамова является полностью гомоморфной системой.

3.2. Построение схемы гомоморфного шифрования

Данная схема гомоморфного шифрования, основанная на построенном гомоморфизме, позволяет вычислять любую полиномиальную функцию $f(A)$, где A - целочисленный вектор $A = \{a_0, a_1, ..., a_n\}, n \in Z$.

Замечание 3 *Если полиномиальная функция f содержит операцию деления, то представления $a_i, i \in [1..n]$ можно считать элементами поля многочленов с рациональными коэффициентами.*

3.2.1. Шифрование и выбор секретного ключа

Шифрование целочисленного вектора $\{a_0, a_1, ..., a_n\}$ происходит согласно алгоритму 3.1, при этом в качестве секретного ключа выступает пара $(r, c(x))$. Таким образом, вычисление функции $f(A)$ сводится к вычислению $f(Gom(A))$.

Замечание 4 *Важно отметить, что если $\exists a_i : a_i < r$, то дальшейшее шифрование не будет иметь никакого смысла. Данный факт является основным недостатком системы.*

3.2.2. Расшифрование

Для восстановления результата необходимо решить уравнение $c(x) = r$, по построению оно будет иметь единственное действительное решение - s_r. Данное решение может быть вычислено на этапе шифрования и в последствии использоваться вместо пары $(r, c(x))$.

3.2.3. Пример

Возьмем в качестве примера два целых числа $a = 197, b = 27$. Положим, что $r = 8$ ($r < min\{27, 197\}$). Переходим в восьмиричную систему счисления: $a = (305)_8, b = (33)_8$.

$$\begin{aligned} a(x) &= 5 + 3x^2 \\ b(x) &= 3 + 3x \end{aligned} \tag{4}$$

Пусть $c(y) = y - 2$, так как $y - 2 = 8$ имеет единственный корень $s_r = 10$, тогда результатом шифрования является:

$$\begin{aligned} c(a)(x) &= 5 + 3(y-2)^2 = 17 - 12y + 3y^2 \\ c(b)(x) &= 3 + 3(y-2) = -3 + 3y \end{aligned} \tag{5}$$

Рассмотрим все арифметические функции.

- Сложение:
  $Sum(y) = 14 - 9y + 3y^2 \Rightarrow Sum(s_r) = 224 = 197 + 27$
- Вычитание:
  $Diff(y) = 20 - 15y + 3y^2 \Rightarrow Diff(s_r) = 170 = 197 - 27$
- Умножение:
  $Mult(y) = 9y^3 - 45y^2 + 87y - 51 \Rightarrow Mult(s_r) = 5319 = 197 * 27$
- Деление:
  $Div(y) = y - 3 + \frac{8}{3y-3} \Rightarrow Div(s_r) = 5 + \frac{8}{27} = 197/27$

3.3. Криптоанализ системы

Рассмотрим применимость некоторых типов атак к системе А. Абрамова.

3.3.1. Атака с подобранным открытым текстом

Предполагаем, что у злоумышленника есть возможность использовать систему: передавать в нее функции для вычисления, исходные данные и получать зашифрованные данные.
Злоумышленник передает целочисленный набор данных $\{a_1, ..., a_n\}$ и получает зашифрованные данные $\{res_1(x), ..., res_n(x)\}$ соответственно. Если $res_i(x) \in Z$, то можно сделать вывод, что $r > a_i$, иначе $r \leq a_i$. В любом случае, $r \in [2, ..., max\{a_i, i \in [1..n]\}]$, предыдущее наблюдение лишь уменьшает перебор возможных оснований для поиска закрытого ключа.

Закрытый ключ можно найти перебором по всем возможным r следующим способом:

1. Переводим $a_i, i \in [1..n]$ в систему счисления с основанием r и перезаписываем представление в виде многочлена $a_i(x), i \in [1..n]$.

2. Решаем задачу декомпозиции $res_i(x) = a_i(x) \circ s(x), i \in [1..n]$, где $s(x)$ - искомый многочлен. Заметим, что res_i, a_i нам известны, значит знаем степень s: $deg(s) = deg(res_i) - deg(a_i) = k, i \in [1..n], k \in Z$. Дальнейшее решение задачи сводится к соотнесению коэффициентов левых и правых частей.

Таким образом, в результате перебора $max\{a_i, i \in [1..n]\}$ вариантов можно найти истинные $r, s(x)$, а значит, в зависимости от размера исходных данных $a_i, i \in [1..n]$ возможно успешное применение данной атаки к системе А. Абрамова.

3.3.2. Атака с подобранным шифротекстом

Допустим, у злоумышленника есть только набор шифротекста $\{res_1, ..., res_n\}$, который был получен преобразованием открытых данных $\{a_1, ..., a_n\}$. Если $\exists m \in [1..n] : res_m \in Z$, тогда $r \geq res_m$, иначе $r \geq 2$. Для нахождения закрытого ключа необходимо решить задачу декомпозиции $res_i(x) = a_i(x) \circ s(x), i \in [1..n]$, где нам известно только $res_i(x)$ и соответственно $deg(res_i) = d_i = deg(a_i) + deg(s)$.

К одному из вариантов решения можно отнести рассмотрение всех возможных степеней $deg(a_i), deg(s)$, зная, что $deg(res_i) = deg(a_i) + deg(s)$, однако данный способ малоэффективен, так как он подразумевает под собой перебор не только степеней, но и коэффициентов многочленов.

Рассмотрим другой возможный подход - если $n \geq 2$, тогда:

1. Находим производные $res'(x), \forall i \in [1..n]$;

2. Для $\forall i, j \in [1..n] : i \neq j$ вычисляем НОД(res_i, res_j);

3. Рекурсивно повторяем второй шаг, до тех пор, пока не останется один элемент. Несложно видеть, что он будет иметь вид $k(x) * s'(x)$;

4. Затем, используя существующие полиномиальные алгоритмы разложения на множители полинома [1] [4], можно найти s', но к сожалению, данные алгоритмы достаточно эффективны только при малых r.

Таким образом, нельзя говорить об успешном применении подобной атаки на систему А. Абрамова. Однако задача доказательства невозможности осуществления этой атаки остается открытой.

4. Реализация схемы гомоморфного шифрования

Для реализации системы гомоморфного шифрования использовался язык Java. Такой выбор во многом был обоснован несколькими причинами:

1. Наличием удобных стандартных библиотек для работы с длинными числами.

2. Несмотря на большое количество доступных библиотек для языка Java, достаточно удобной библиотеки для работы с многочленами, которая может использоваться при гомоморфном шифровании, не было.

4.1. Реализация библиотеки многочленов

Библиотека для работы с многочленами состоит из следующих основных классов:

1. Класс Monom (одночлен) содержит методы для работы с одночленами (сложение, вычитание, умножение, деление).

2. Класс Polynom (многочлен) содержит список одночленов и реализует методы для работы с ними. В публичный интерфейс вынесены операции сложения, вычитания, деления, возведения в степень, композиции многочленов.

3. Класс PolynomFraction (дробный многочлен) отвечает за работу с дробными многочленами, также доступны операции сложения, вычитания, деления, умножения. Для нахождения НОК и НОД при операциях сложения и вычитания используются методы вспомогательного класса утилит.

4. Класс RationalPolynom (рациональный полином) расширение класса Polynom, помимо целой части добавляется дробная. Доступны

стандартные арифметические операции, методы для выделения целой части.

5. Класс PolynomUtils предоставляет вспомогательные операции для работы с многочленами, такие как: генерация случайного полинома по заданной степени, разложение числа в полином по заданному основанию системы счисления, нахождение НОК и НОД для многочленов.

Все публичные методы классов библиотеки покрыты тестами с использованием тестового фреймворка JUnit.

4.2. Реализация схемы гомоморфного шифрования

Главными этапами работы рассматриваемой схемы являются генерация ключа, шифрование и дешифрование. Соответственно клиентский код для работы с данной схемой разбит на три основных класса.

1. Класс SecretKey.
   Данный класс инкапсулирует логику для работы с секретным ключом, в публичный интерфейс вынесен статический метод генерации экземпляра данного класса. Входные параметры – степень генерируемого многочлена и основание системы счисления, используемое при генерации. На выходе получаем секретный ключ, состоящий из многочлена и его корня.

2. Класс Encryption.
   Содержит набор методов для шифрования входной последовательности чисел. Базовый метод принимает на вход число в формате BigInteger. В самом методе происходит формирование случайного основания системы счисления и генерация случайного многочлена. Метод возвращает экземпляр класса Polynom, который является результатом шифрования.

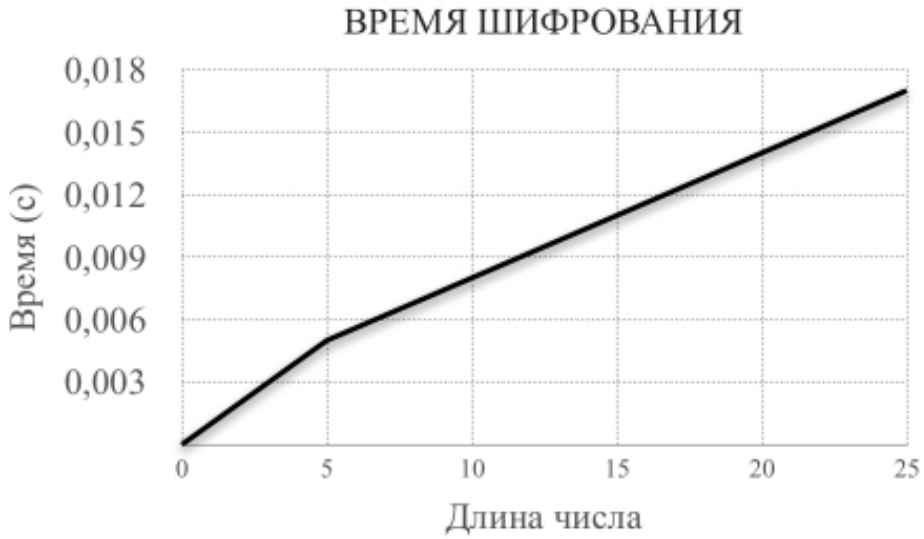
3. Класс Decryption.
   Для операции дешифрования используется поле экземпляра клас-

са SecretKey – корень многочлена, вычисленное на этапе формирования секретного ключа. Таким образом в данной классе присутствует один метод, который из значений объектов SecretKey и Polynom вычисляет исходное зашифрованное число.

5. Характеристики работы системы

В рамках данной работы были проведены замеры производительности реализованной системы. Производительность оценивалась на компьютере с ОС Windows 7 Enterprise, четырехъядерным процессором Intel Core i5-5300U 2.30GHz и оперативной памятью 12Гб.

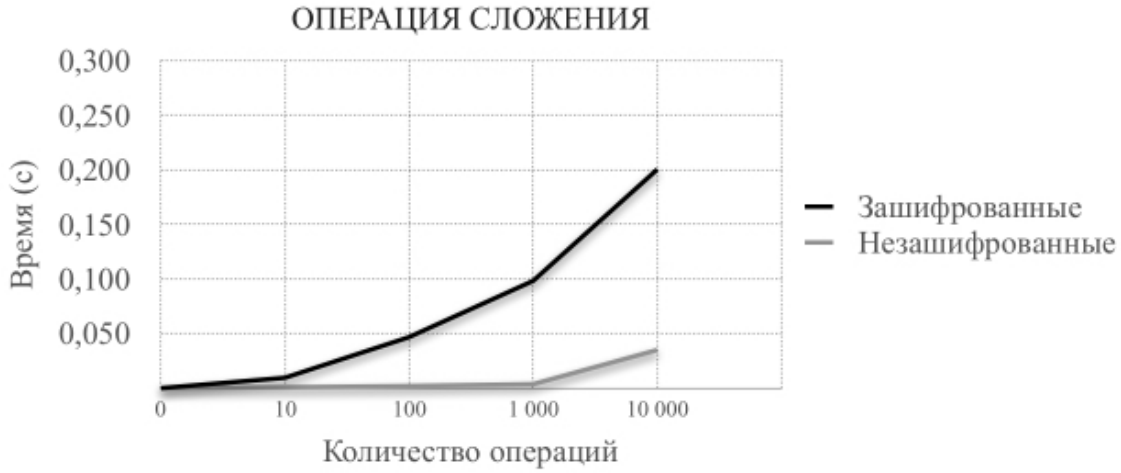
Была определена зависимость времени выполнения операции шифрования от длины открытых данных. Основания системы счисления и степени многочлена при шифровании были случайными. На графике ниже отражен получившийся результат. Как можно наблюдать, зависимость является линейной, что является приемлемым для применения в реальных системах.



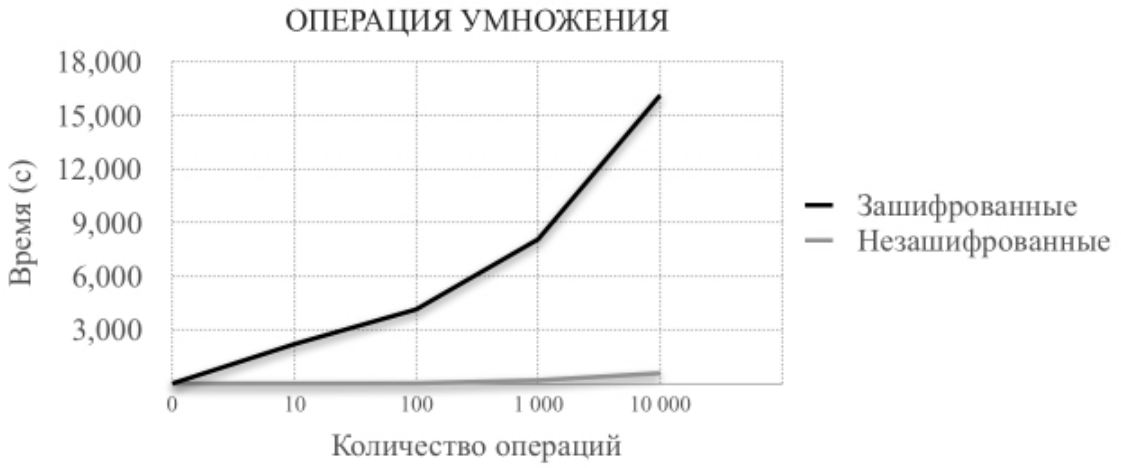
Далее рассматривалось сравнение операций сложения и умножения над открытыми и зашифрованными данными. Для облегчения тестирования данные шифровались с одинаковым, заранее заданным секретным ключом.

На следующем графике показано сравнение операций сложения. Классические алгоритмы выполняют сложение многочленов за $O(n)$, что и подтверждается полученными результатами - наблюдается линейный рост времени при увеличении количества операций над зашифрованными данными, представленными в виде многочленов. Это объясняется

увеличением количества выделяемой памяти для коэффициентов многочленов и повышением сложности операции сложения над числами большей размерности.



Ниже представлено аналогичное сравнение для операций умножения. В отличие от операции сложения, умножение многочленов имеет квадратичное время выполнения $O(n^2)$ - при увеличении количества операций можно наблюдать сильный рост времени выполнения, что является одним из главных недостатков рассматриваемой системы.



Заключение

В данной работе рассматривались полностью гомоморфные схемы шифрования, исследования в области которых активно ведутся в последнее десятелетие в связи с бурным развитием облачных технологий. Была реализована одна из подобных схем шифрования - схема, предложенная А. Абрамовым. В процессе изучения данной схемы, основанной на гомоморфизме в кольце многочленов, обнаружились следующие недостатки существующих решений - степени многочленов при операциях умножения растут слишком быстро, делая невозможным использование схемы на практике; потенциальные уязвимости к некоторым видам атак. Несмотря на эти проблемы, данная область по прежнему видится перспективной и есть основания полагать, что исследования будут продолжаться. Таким образом, разработанная в работе библиотека для работы с многочленами послужит хорошим базисом для дальнейшего изучения и улучшения существующей схемы.

К результатам выпускной работы относится:

1. Реализация библиотеки для работы с целыми и дробными многочленами.

2. Реализация схемы полностью гомоморфного шифрования, предложенной А. Абрамовым.

3. Тестирование реализации схемы гомоморфного шифрования.

Реализация находится по адресу:
https://github.com/liza-efremova/fully-homomorphic-encryption.

Список литературы

[1] Darel W. Hardy Carol L. Walker. Applied Algebra: Codes, Ciphers, and Discrete Algorithms. — Prentice Hall, 2003. — 420 p.

[2] Gentry Craig. A fully homomorphic encryption scheme // PhD thesis. – Stanford University. — 2009. — URL: http://crypto.stanford.edu/craig.

[3] Gentry Craig Halevi Shai Smart Nigel P. Homomorphic Evaluation of the AES Circuit. // Lecture Notes in Computer Science. — 2012. — Vol. 7417. — P. 850–867. — URL: https://eprint.iacr.org/2012/099.pdf.

[4] Lenstra A. K.; Lenstra H. W.; Lovász László. Factoring polynomials with rational coefficients // Mathematische Annalen. — 1982. — Vol. 261, no. 4. — P. 515–534.

[5] R.L. Rivest L. Adleman, Dertouzos M.L. On data banks and privacy homomorphisms. In Foundations of Secure Computation // Foundations of secure computation. — 1978. — Vol. 32, no. 4. — P. 169–178. — URL: http://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/RAD78.pdf.

[6] А. О. Жиров О. В. Жирова С. Ф. Кренделев. Безопасные облачные вычисления с помощью гомоморфной криптографии // НГУ-Parallels. — 2011. — URL: http://bit.mephi.ru/wp-content/uploads/2013/2013_1/part_1.pdf.

[7] Абрамов А. Гомоморфное шифрование. — 2015.

[8] Введение в криптографию / В. В. Ященко, Н. П. Варновский, Ю. В. Нестеренко и др. ; Под ред. В. В. Ященко. — М. : МЦНМО, 2012. — 348 с.